

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-311821

(43)公開日 平成9年(1997)12月2日

(51)Int.Cl.<sup>5</sup>

G 0 6 F 12/14

識別記号

3 1 0

庁内整理番号

F I

G 0 6 F 12/14

技術表示箇所

3 1 0 K

3 1 0 A

審査請求 未請求 請求項の数6 O L (全 18 頁)

(21)出願番号

特願平8-128635

(22)出願日

平成8年(1996)5月23日

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 古林 三郎

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

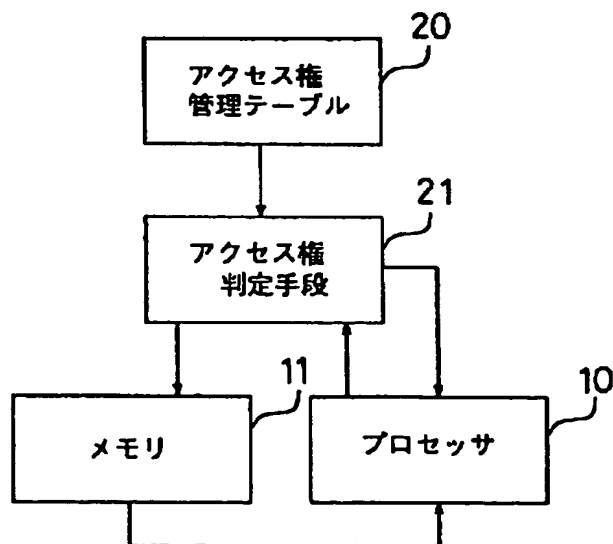
(74)代理人 弁理士 田澤 博昭 (外2名)

(54)【発明の名称】 記憶データ保護装置

(57)【要約】

【課題】 記憶データの保護が、オペレーティングシステムにおける制御の基本単位であるプロセス間でしか行えなかった。

【解決手段】 メモリ11上のアドレス範囲で定義される各種データセグメント毎にアクセスが許可されているメモリ11上のモジュールのアドレス範囲を格納しているアクセス権管理テーブル20と、プロセッサ10から要求されたアドレスがアクセス権管理テーブル20で定義されたデータセグメントである場合にそのアクセス権管理テーブル20を用いて現在のプログラムカウンタに対応するモジュールがそのデータセグメントにアクセス可能であるかどうかを判定するアクセス権判定手段21とを備えた。



## 1

## 【特許請求の範囲】

【請求項 1】 プロセッサおよびメモリを有する電子計算機における単一のプログラム内の各種機能モジュール間で記憶データの保護を行う記憶データ保護装置において、上記メモリ上のアドレス範囲で定義される各種データセグメント毎にアクセスが許可されているメモリ上のモジュールのアドレス範囲を格納するアクセス権管理テーブルと、上記プロセッサから要求されたアドレスが上記アクセス権管理テーブルで定義されたデータセグメントである場合にそのアクセス権管理テーブルを用いて現在のプログラムカウンタに対応するモジュールがそのデータセグメントをアクセス可能であるかどうかを判定するアクセス権判定手段とを備えたことを特徴とする記憶データ保護装置。

【請求項 2】 アクセス権管理テーブルは、メモリ上のアドレス範囲で各種モジュールを定義するモジュール管理テーブルと、上記メモリ上のアドレス範囲で定義される各種データセグメント毎にアクセスが許可されているモジュールを上記モジュール管理テーブル上のインデックスで複数登録するデータセグメント管理テーブルとを備えたことを特徴とする請求項 1 記載の記憶データ保護装置。

【請求項 3】 アクセス権管理テーブルを更新するための空き領域があるかを検査し、空き領域が不足と判定された場合にそのアクセス権管理テーブルの拡張領域を確保するアクセス権管理テーブル更新手段を備えたことを特徴とする請求項 2 記載の記憶データ保護装置。

【請求項 4】 アクセス権管理テーブル更新手段を用いてモジュール管理テーブルにモジュールを登録するモジュールインストール手段と、メモリの獲得または解放に合わせてデータセグメント管理テーブルにデータセグメントを登録または削除するメモリ獲得解放手段と、アクセス許可を申請したモジュールを上記データセグメント管理テーブルのアクセス可能モジュールに追加するデータアクセス宣言手段とを備えたことを特徴とする請求項 3 記載の記憶データ保護装置。

【請求項 5】 アクセス権判定手段によって検出された不当アクセスによるプロセッサからの通知を検出するトラップ検出手段と、そのトラップ検出手段がトラップを検出した際に不当アクセスの内容としてアクセス権管理テーブルの内容とプログラムカウンタとアクセスアドレスを二次記憶装置に記録する不当アクセス内容記録手段とを備えたことを特徴とする請求項 4 記載の記憶データ保護装置。

【請求項 6】 各モジュール毎に切り離しが可能か否かを表すフラグと切り離し手続き用の関数アドレスを有するモジュール管理テーブルと、トラップ検出手段からの要求に応じて上記フラグを検査し、切り離しが可能であれば上記切り離し手続き用の関数アドレスから始まる関数を実行し、不当アクセスを行ったモジュールの切り離

## 2

しを行うモジュールアンインストール手段とを備えたことを特徴とする請求項 5 記載の記憶データ保護装置。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 この発明は、プロセッサおよびメモリを有する電子計算機において、単一のプログラム内の各種機能モジュール間で記憶データの保護を行う記憶データ保護装置に関するものである。

## 【0002】

【従来の技術】 図 21 は例えば特開平 4-199341 号公報に示された従来の記憶データ保護装置を示すブロック図であり、図において、10 はプロセッサ、11 はメモリ、12 はメモリ 11 上のアドレス範囲で定義される各種データセグメント毎にアクセス可能なプロセスを示すメモリアccessキーを格納するアクセス権管理テーブルである。13 は現在走行するプロセスのメモリアccessキーを保持するアクセスキーレジスタ、14 はアクセス権管理テーブル 12 とアクセスキーレジスタ 13 を用いて現在走行するプロセスがプロセッサ 10 から要求されたアドレスに対応するデータセグメントにアクセス可能であるかどうかを判定するアクセス権判定手段である。アクセスキーレジスタ 13 には、プロセスが走行を開始する時に対応するメモリアccessキーが設定される。

【0003】 図 22 はアクセス権管理テーブルの構造を示す説明図であり、図において、15 はアクセス権管理テーブル 12 のインデックスとしてのデータセグメントインデックス、16 はデータセグメント開始アドレス、17 はデータセグメントサイズ、18 はデータセグメントインデックス 15 で示されるデータセグメントに対してアクセス可能なプロセスのメモリアccessキーである。

【0004】 次に動作について説明する。図 23 は従来の記憶データ保護装置におけるアクセス権判定処理の流れを示すフローチャートであり、この図 23 を用いて説明する。まず、ステップ ST231 でプロセッサ 10 がアクセスしたいアドレスをアクセス権判定手段 14 に送ると、アクセス権判定手段 14 は、ステップ ST232 で送られたアドレスをもとにアクセス権管理テーブル 12 を参照し、アドレスに対応したメモリアccessキー 18 を取得する。

【0005】 次に、ステップ ST233 で、取得したメモリアccessキー 18 をアクセスキーレジスタ 13 の内容と比較し、現在走行しているプロセスに対してアクセスが許可されているかどうかの判定を行う。アクセスが許可されていると判定した場合は、ステップ ST234 に進み、アドレスをメモリ 11 に送る。そして、ステップ ST235 で、メモリ 11 はアドレスに対応したメモリ 11 の内容をプロセッサ 10 に送る。ステップ ST233 にてアクセスが許可されていないと判定した場合

## 3

は、ステップST236に進み、プロセッサ10に対し不当アクセス要求であることを通知する。

## 【0006】

【発明が解決しようとする課題】従来の記憶データ保護装置は以上のように構成されているので、記憶データの保護が、オペレーティングシステムにおける制御の基本単位であるプロセス間でしか行うことができなかった。プロセス、即ちプログラムは複数の機能モジュールの集合体である。例えば、複数の開発者が作成した異なる機能モジュールを組み上げてできるプログラムにおいて、ある機能モジュールが全く関係のない他の機能モジュールのデータを破壊してしまったような場合、従来技術による記憶データ保護装置は効力を持つことができない。この場合、不当なアクセスにも拘らずその不当なアクセスを検出することができないため、プログラムの暴走を引き起こしてしまうなどの課題があった。

【0007】この発明は上記のような課題を解決するためになされたもので、1つのプログラム内の複数の機能モジュール間で記憶データの保護を実現することができる記憶データ保護装置を得ることを目的とする。

## 【0008】

【課題を解決するための手段】請求項1記載の発明に係る記憶データ保護装置は、メモリ上のアドレス範囲で定義される各種データセグメント毎にアクセスが許可されているメモリ上のモジュールのアドレス範囲を格納するアクセス権管理テーブルと、プロセッサから要求されたアドレスが上記アクセス権管理テーブルで定義されたデータセグメントである場合にそのアクセス権管理テーブルを用いて現在のプログラムカウンタに対応するモジュールがそのデータセグメントをアクセス可能であるかどうかを判定するアクセス権判定手段とを備えたものである。

【0009】請求項2記載の発明に係る記憶データ保護装置は、アクセス権管理テーブルに、メモリ上のアドレス範囲で各種モジュールを定義するモジュール管理テーブルと、上記メモリ上のアドレス範囲で定義される各種データセグメント毎にアクセスが許可されているモジュールを上記モジュール管理テーブル上のインデックスで複数登録するデータセグメント管理テーブルとを備えたものである。

【0010】請求項3記載の発明に係る記憶データ保護装置は、アクセス権管理テーブルを更新するための空き領域があるかを検査し、空き領域が不足と判定された場合にそのアクセス権管理テーブルの拡張領域を確保するアクセス権管理テーブル更新手段を備えたものである。

【0011】請求項4記載の発明に係る記憶データ保護装置は、アクセス権管理テーブル更新手段を用いてモジュール管理テーブルにモジュールを登録するモジュールインストール手段と、メモリの獲得または解放に合わせてデータセグメント管理テーブルにデータセグメントを

## 4

登録または削除するメモリ獲得解放手段と、アクセス許可を申請したモジュールを上記データセグメント管理テーブルのアクセス可能モジュールに追加するデータアクセス宣言手段とを備えたものである。

【0012】請求項5記載の発明に係る記憶データ保護装置は、アクセス権判定手段によって検出された不当アクセスによるプロセッサからの通知を検出するトラップ検出手段と、そのトラップ検出手段がトラップを検出した際に不当アクセスの内容としてアクセス権管理テーブルの内容とプログラムカウンタとアクセスアドレスを二次記憶装置に記録する不当アクセス内容記録手段とを備えたものである。

【0013】請求項6記載の発明に係る記憶データ保護装置は、各モジュール毎に切り離しが可能か否かを表すフラグと切り離し手続き用の関数アドレスを有するモジュール管理テーブルと、トラップ検出手段からの要求に応じて上記フラグを検査し、切り離しが可能であれば上記切り離し手続き用の関数アドレスから始まる関数を実行し、不当アクセスを行ったモジュールの切り離しを行うモジュールアンインストール手段とを備えたものである。

## 【0014】

【発明の実施の形態】以下、この発明の実施の一形態を説明する。

実施の形態1. 図1はこの発明の実施の形態1による記憶データ保護装置を示すブロック図であり、図において、10はプロセッサ、11はメモリ、20はメモリ11上のアドレス範囲で定義される各種データセグメント毎にアクセスが許可されているメモリ11上のモジュールのアドレス範囲を格納しているアクセス権管理テーブル、21はプロセッサ10から要求されたアドレスがアクセス権管理テーブル20で定義されたデータセグメントである場合にアクセス権管理テーブル20を用いて現在のプログラムカウンタに対応するモジュールがそのデータセグメントをアクセス可能であるかどうかを判定するアクセス権判定手段である。

【0015】図2はアクセス権管理テーブルの構造を示す説明図であり、図において、15はアクセス権管理テーブル20のインデックスとしてのデータセグメントインデックス、16はデータセグメント開始アドレス、17はデータセグメントサイズである。22は各データセグメントインデックス15に対応するデータセグメントについてのアクセス可能モジュール開始アドレス、23はアクセス可能モジュールサイズである。

【0016】次に動作について説明する。図3はこの実施の形態1による記憶データ保護装置におけるアクセス権判定処理の流れを示すフローチャートであり、以下、この図3を参照しながら説明する。まず、ステップST31でプロセッサ10がアクセスしたいアドレスをアクセス権判定手段21に送ると、アクセス権判定手段21

## 5

は、ステップST32で送られたアドレスをもとにアクセス権管理テーブル20のデータセグメント開始アドレス16とデータセグメントサイズ17を参照し、送られたアドレスに対応するデータセグメントインデックス15を取得する。

【0017】次に、ステップST33で、現在のプログラムカウンタが取得したモジュールと、送られたアドレスに対応するデータセグメントインデックス15のアクセス可能モジュール開始アドレス22及びアクセス可能モジュールサイズ23から、現在のプログラムカウンタが取得したモジュールがアクセス可能モジュールの範囲内であるかどうかを検査して、アクセスが許可されているかどうか判定を行う。アクセスが許可されていると判定した場合は、ステップST34に進み、アドレスをメモリ11に送る。そして、ステップST35で、メモリ11はアドレスに対応したメモリ11の内容をプロセッサ10に送る。ステップST33にてアクセスが許可されていないと判定した場合は、ステップST36に進み、プロセッサ10に対して不当アクセス要求であることを通知する。

【0018】以上のように、この実施の形態1によれば、現在のプログラムカウンタをもとにアクセス権管理テーブル20を参照して、モジュール単位で記憶データの保護を行うことができるため、1つのプログラム内の複数の機能モジュール間で記憶データの保護を実現することができる効果がある。

【0019】実施の形態2。図4はこの発明の実施の形態2による記憶データ保護装置を示すブロック図であり、図において、30はメモリ11上のアドレス範囲で各種モジュールを定義するモジュール管理テーブル、31はメモリ11上のアドレス範囲で定義される各種データセグメント毎にアクセスが許可されているモジュールをモジュール管理テーブル30上のインデックスで複数登録されたデータセグメント管理テーブルである。

【0020】図5はモジュール管理テーブルの構造を示す説明図であり、図において、32はモジュール管理テーブル30のインデックスとしてのモジュールインデックス、33はモジュール開始アドレス、34はモジュールサイズである。図6はデータセグメント管理テーブルの構造を示す説明図であり、図において、35はデータセグメントインデックス15に対応するデータセグメントに対してアクセスが許可されたモジュールをモジュールインデックス32で示すアクセス可能モジュールインデックスである。なお、その他の構成については上記実施の形態1と同一なのでその重複する説明を省略する。

【0021】次に動作について説明する。図7はこの実施の形態2による記憶データ保護装置におけるアクセス権判定処理の流れを示すフローチャートであり、以下、この図7を参照しながら説明する。まず、ステップST71でプロセッサ10がアクセスしたいアドレスをアク

## 6

セス権判定手段21に送ると、アクセス権判定手段21は、ステップST72では送られたアドレスをもとにデータセグメント管理テーブル31のデータセグメント開始アドレス16とデータセグメントサイズ17を参照し、送られたアドレスに対応するデータセグメントインデックス15を取得する。

【0022】次に、ステップST73で、取得したデータセグメントインデックス15に対応するアクセス可能モジュールインデックス35を順次取得する。そして、ステップST74にて、モジュール管理テーブル30において、現在のプログラムカウンタが取得したモジュールと、アクセス可能モジュールインデックス35によって順次取得されたモジュールインデックス32のモジュール開始アドレス33及びモジュールサイズ34から、現在のプログラムカウンタが取得したモジュールがアクセス可能モジュールの範囲内であるかどうかを検査して、アクセスが許可されているかどうかの判定を行う。アクセスが許可されていると判定した場合は、ステップST75に進み、アドレスをメモリ11に送る。そして、ステップST76で、メモリ11はアドレスに対応したメモリ11の内容をプロセッサ10に送る。

【0023】ステップST74にてアクセスが許可されていないと判定した場合は、ステップST77に進み、全てのアクセス可能モジュールインデックス35について検査が終了していなかった場合、再びステップST73に戻ってアクセス権判定を行う。ステップST77にて全てのアクセス可能モジュールインデックス35について検査が終了していた場合、ステップST78に進み、プロセッサ10に対して不当アクセス要求であることを通知する。

【0024】以上のように、この実施の形態2によれば、アクセス権判定手段21がアクセスしたデータセグメントについて、アクセス可能モジュールインデックス35として登録されているすべてのモジュールを検査するため、複数のモジュール間で1つのデータセグメントを共有することができる効果がある。

【0025】実施の形態3。図8はこの発明の実施の形態3による記憶データ保護装置を示すブロック図であり、図において、40はアクセス権管理テーブル20を更新するための空き領域があるかを検査し、空き領域が不足と判定された場合にそのアクセス権管理テーブル20の拡張領域を確保するアクセス権管理テーブル更新手段である。なお、その他の構成については上記実施の形態2と同一なのでその重複する説明を省略する。

【0026】次に動作について説明する。図9はこの実施の形態3による記憶データ保護装置におけるアクセス権管理テーブル更新処理の流れを示すフローチャートであり、以下、この図9を参照しながら説明する。アクセス権管理テーブル更新手段40は、まず、ステップST91で、初期設定であった場合にステップST92に進

## 7

み、アクセス権管理テーブル20用の領域を確保する。次にステップST93で、モジュール管理テーブル30におけるモジュールインデックス32が0のエントリにアクセス権管理テーブル更新モジュール40用のエントリを作成する。そしてステップST94で、データセグメント管理テーブル31におけるデータセグメントインデックス15が0のエントリにアクセス権管理テーブル20用のエントリを作成する。

【0027】次にステップST95にて、アクセス権管理テーブル20を更新するための十分な空き領域があるかないかを検査し、空き領域が十分であると判定された場合、ステップST96に進み、アクセス権判定手段21を用いたメモリアクセスにより要求された更新処理を行う。ステップST91にて初期設定でなかった場合はステップST95に進み、アクセス権管理テーブル20を更新するための十分な空き領域があるかないかを検査する。ステップST95にて空き領域が不足と判定された場合、ステップST97に進み、アクセス権管理テーブル20用の拡張領域を確保する。次にステップST98で、データセグメント管理テーブル31の空きエントリにアクセス権管理テーブル20用の追加エントリを作成する。そして、ステップST96で、アクセス権判定手段21を用いたメモリアクセスにより要求された更新処理を行う。

【0028】以上のように、この実施の形態3によれば、アクセス権管理テーブル20を更新できるのはアクセス権管理テーブル更新手段40だけとなるため、アクセス権管理テーブル20を誤操作による破壊から守り、安全に更新することができる効果がある。

【0029】実施の形態4、図10はこの発明の実施の形態4による記憶データ保護装置を示すブロック図であり、図において、50はモジュール管理テーブル30にモジュールを登録するモジュールインストール手段、51はメモリの獲得または解放に合わせてデータセグメント管理テーブル31にデータセグメントを登録したり削除したりするメモリ獲得解放手段、52はアクセス許可を申請したモジュールをデータセグメント管理テーブル31のアクセス可能モジュールインデックス35に追加するデータアクセス宣言手段である。なお、その他の構成については上記実施の形態3と同一なのでその重複する説明を省略する。

【0030】次に動作について説明する。図11はこの実施の形態4による記憶データ保護装置におけるモジュールインストール処理の流れを示すフローチャートであり、以下、この図11を参照しながら説明する。モジュールインストール手段50は、まず、ステップST111で、要求されたモジュールをインストールする。そして、ステップST112で、アクセス権管理テーブル更新手段40を用いてモジュール管理テーブル30にインストールしたモジュールのエントリを追加する。

## 8

【0031】図12はメモリ獲得解放処理の流れを示すフローチャートであり、以下、この図12を参照しながら説明する。メモリ獲得解放手段51は、まず、ステップST121で、要求がメモリ獲得か解放かを判別し、獲得要求の場合にステップST122に進み、アクセス権管理テーブル更新手段40を用いてデータセグメント管理テーブル31に獲得したデータセグメント用のエントリを作成する。次にステップST123で、モジュール管理テーブル30と現在のプログラムカウンタから要求元モジュールに対応するモジュールインデックス32を取得する。

【0032】そして、ステップST124で、アクセス権管理テーブル更新手段40を用いて取得したモジュールインデックス32をアクセス可能モジュールインデックス35としてデータセグメント管理テーブル31に登録する。ステップST121にて、解放要求であった場合、ステップST125に進み、アクセス権管理テーブル更新手段40を用いてデータセグメント管理テーブル31から対応するデータセグメントのエントリを削除する。

【0033】図13はデータアクセス宣言処理の流れを示すフローチャートであり、以下、この図13を参照しながら説明する。データアクセス宣言手段52は、まず、ステップST131で、データセグメント管理テーブル31から宣言されたアドレスに対応するデータセグメントを特定する。次にステップST132で、モジュール管理テーブル30と現在のプログラムカウンタから要求元モジュールに対応するモジュールインデックス32を取得する。

【0034】そして、ステップST133で、取得したモジュールインデックス32が、アクセス宣言されたデータセグメントに対して既にアクセス許可されているかどうかを検査し、まだ許可されていなかった場合は、ステップST134に進み、アクセス権管理テーブル更新手段40を用いて取得したモジュールインデックス32をアクセス可能モジュールインデックス35としてデータセグメント管理テーブル31に登録する。ステップST133にて既に許可されていた場合は、そのまま処理を終了する。

【0035】以上のように、この実施の形態4によれば、モジュールインストール手段50がモジュールインストール時にモジュール管理テーブル30を作成し、メモリ獲得解放手段51がデータセグメントの生成または削除に合わせてデータセグメント管理テーブル31の更新を行い、データアクセス宣言手段52がモジュールが自発的に宣言した時にアクセス可能モジュールインデックス35として上記モジュールをデータセグメント管理テーブル31に登録するため、動的に生成及び削除されるようなデータに対するアクセス権管理テーブル20の更新処理を自動的に行うことができる効果がある。

【0036】実施の形態5. 図14はこの発明の実施の形態5による記憶データ保護装置を示すブロック図であり、図において、60は二次記憶装置、61はアクセス権判定手段21によって検出された不当アクセスによるプロセッサ10からの通知を検出するトラップ検出手段、62はトラップを検出した際に不当アクセスの内容としてアクセス権管理テーブル20の内容とプログラムカウンタとアクセスアドレスを二次記憶装置60に記録する不当アクセス内容記録手段である。なお、その他の構成については上記実施の形態4と同一なのでその重複する説明を省略する。

【0037】次に動作について説明する。図15はこの実施の形態5による記憶データ保護装置におけるトラップ検出処理の流れを示すフローチャートであり、以下、この図15を参照しながら説明する。トラップ検出手段61は、まず、ステップST151で、プロセッサ10からの不当アクセスの通知をトラップとして検出する。次にステップST152で、不当アクセス内容記録手段62を用いて不当アクセスの内容を二次記憶装置60上に記録する。そして、ステップST153で、アクセス要求元に対して不当アクセスを通知する。

【0038】図16は不当アクセス内容記録処理の流れを示すフローチャートであり、以下、この図16を参照しながら説明する。不当アクセス内容記録手段62は、まず、ステップST161で、現在時刻を取得する。次にステップST162で、二次記憶装置60上に取得した現在時刻をタグとする領域を確保する。そして、ステップST163で、確保した不当アクセス内容記録領域にアクセス権管理テーブル20の内容を記録し、続くステップST164で、上記領域にトラップ検出時のプログラムカウンタとアクセスアドレスを記録する。

【0039】以上のように、この実施の形態5によれば、プロセッサ10からの不当アクセスの通知を検出したトラップ検出手段61が、不当アクセス内容記録手段62を用いて不当アクセスの内容を二次記憶装置60に記録するため、不当アクセスの内容を確実に且つ容易に知ることができる効果がある。

【0040】実施の形態6. 図17はこの発明の実施の形態6による記憶データ保護装置を示すブロック図であり、図において、70はトラップ検出手段61からの要求に応じて可能であれば不当アクセスを行ったモジュールの切り離しを行うモジュールアンインストール手段である。図18はモジュール管理テーブルの構造を示す説明図であり、図において、71は対応するモジュールが切り離し可能か否かを表す切り離し可能フラグ、72は対応するモジュールの接続状態、73は対応するモジュールの切り離し手続き関数アドレスである。なお、その他の構成については上記実施の形態5と同一なのでその重複する説明を省略する。

【0041】次に動作について説明する。図19はこの

実施の形態6による記憶データ保護装置におけるトラップ検出処理の流れを示すフローチャートであり、以下、この図19を参照しながら説明する。トラップ検出手段61は、まず、ステップST191で、プロセッサ10からの不当アクセスの通知をトラップとして検出する。次にステップST192で、不当アクセス内容記録手段62を用いて不当アクセスの内容を二次記憶装置60上に記録する。そして、ステップST193で、モジュールアンインストール手段70を用いて可能であれば不当アクセスを要求したモジュールを切り離す。そして、ステップST194で、切り離しが失敗した場合、ステップST195に進み、アクセス要求元に対して不当アクセスを通知する。ステップST194にて、切り離しが成功した場合、そのまま処理を終了する。

【0042】図20はモジュールアンインストール処理の流れを示すフローチャートであり、以下、この図20を参照しながら説明する。モジュールアンインストール手段70は、まず、ステップST201で、モジュール管理テーブル30とトラップ検出時のプログラムカウンタから不当アクセスを行ったモジュールを特定する。次にステップST202で、モジュール管理テーブル30の切り離し可能フラグ71から特定したモジュールが切り離し可能か否かをチェックする。切り離し可能であった場合にステップST203に進み、モジュール管理テーブル30に登録されている切り離し手続き関数アドレス73から始まる関数を実行する。そして、ステップST204で、切り離しが成功した場合、ステップST205に進み、アクセス権管理テーブル更新手段40を用いてモジュール管理テーブル30内の接続状態72をアンインストールにする。

【0043】接続状態72がアンインストールのモジュール管理テーブル30のエントリは削除されたものと見なす。そして、ステップST206で、要求元であるトラップ検出手段61に切り離し成功と通知する。ステップST202にて、切り離しが不可能であった場合はステップST207に進み、要求元であるトラップ検出手段61に切り離し失敗と通知する。ステップST204にて切り離しが失敗した場合もステップST207に進み、要求元であるトラップ検出手段61に切り離し失敗と通知する。

【0044】以上のように、この実施の形態6によれば、モジュール管理テーブル30に各モジュール毎に切り離しが可能かどうか定義されており、トラップ検出手段61がモジュールアンインストール手段70を用いて、可能であれば不当アクセスを行ったモジュールの切り離しを自動的に行うため、プログラムの実行をできる限り継続させることができる効果がある。

【0045】

【発明の効果】以上のように、請求項1記載の発明によれば、プロセッサから要求されたアドレスがアクセス権

管理テーブルで定義されたデータセグメントである場合にそのアクセス権管理テーブルを用いて現在のプログラムカウンタに対応するモジュールがそのデータセグメントをアクセス可能であるかどうかを判定するアクセス権判定手段を備えるように構成したので、現在のプログラムカウンタをもとにアクセス権管理テーブルを参照して、モジュール単位で記憶データの保護を行うことができるため、1つのプログラム内の複数の機能モジュール間で記憶データの保護を実現することができる効果がある。

【0046】請求項2記載の発明によれば、アクセス権管理テーブルに、メモリ上のアドレス範囲で各種モジュールを定義するモジュール管理テーブルと、メモリ上のアドレス範囲で定義される各種データセグメント毎にアクセスが許可されているモジュールをモジュール管理テーブル上のインデックスで複数登録するデータセグメント管理テーブルとを備えるように構成したので、アクセス権判定手段がアクセスしたデータセグメントについて、アクセス可能モジュールインデックスとして登録されているすべてのモジュールを検査するため、複数のモジュール間で1つのデータセグメントを共有することができる効果がある。

【0047】請求項3記載の発明によれば、アクセス権管理テーブルを更新するための空き領域があるかを検査し、空き領域が不足と判定された場合にそのアクセス権管理テーブルの拡張領域を確保するアクセス権管理テーブル更新手段を備えるように構成したので、アクセス権管理テーブルを更新できるのはアクセス権管理テーブル更新手段だけとなるため、アクセス権管理テーブルを誤操作による破壊から守り、安全に更新することができる効果がある。

【0048】請求項4記載の発明によれば、アクセス権管理テーブル更新手段を用いてモジュール管理テーブルにモジュールを登録するモジュールインストール手段と、メモリの獲得または解放に合わせてデータセグメント管理テーブルにデータセグメントを登録または削除するメモリ獲得解放手段と、アクセス許可を申請したモジュールをデータセグメント管理テーブルのアクセス可能モジュールに追加するデータアクセス宣言手段とを備えるように構成したので、モジュールインストール手段がモジュールインストール時にモジュール管理テーブルを作成し、メモリ獲得解放手段がデータセグメントの生成または削除に合わせてデータセグメント管理テーブルの更新を行い、データアクセス宣言手段がモジュールが自動的に宣言した時にアクセス可能モジュールインデックスとして上記モジュールをデータセグメント管理テーブルに登録するため、動的に生成及び削除されるようなデータに対するアクセス権管理テーブルの更新処理を自動的に行うことができる効果がある。

【0049】請求項5記載の発明によれば、アクセス権

判定手段によって検出された不当アクセスによるプロセッサからの通知を検出するトラップ検出手段と、トラップを検出した際に不当アクセスの内容としてアクセス権管理テーブルの内容とプログラムカウンタとアクセスアドレスを二次記憶装置に記録する不当アクセス内容記録手段とを備えるように構成したので、プロセッサからの不当アクセスの通知を検出したトラップ検出手段が、不当アクセス内容記録手段を用いて不当アクセスの内容を二次記憶装置に記録するため、不当アクセスの内容を確実に且つ容易に知ることができる効果がある。

【0050】請求項6記載の発明によれば、各モジュール毎に切り離しが可能か否かを表すフラグと切り離し手続き用の関数アドレスを有するモジュール管理テーブルと、トラップ検出手段からの要求に応じてフラグを検査し、切り離しが可能であれば切り離し手続き用の関数アドレスから始まる関数を実行し、不当アクセスを行ったモジュールの切り離しを行うモジュールアンインストール手段とを備えるように構成したので、モジュール管理テーブルに各モジュール毎に切り離しが可能かどうかで定義されており、トラップ検出手段がモジュールアンインストール手段を用いて、可能であれば不当アクセスを行ったモジュールの切り離しを自動的に行うため、プログラムの実行をできる限り継続させることができる効果がある。

#### 【図面の簡単な説明】

【図1】 この発明の実施の形態1による記憶データ保護装置を示すブロック図である。

【図2】 アクセス権管理テーブルの構造を示す説明図である。

【図3】 実施の形態1による記憶データ保護装置におけるアクセス権判定処理の流れを示すフローチャートである。

【図4】 この発明の実施の形態2による記憶データ保護装置を示すブロック図である。

【図5】 モジュール管理テーブルの構造を示す説明図である。

【図6】 データセグメント管理テーブルの構造を示す説明図である。

【図7】 実施の形態2による記憶データ保護装置におけるアクセス権判定処理の流れを示すフローチャートである。

【図8】 この発明の実施の形態3による記憶データ保護装置を示すブロック図である。

【図9】 実施の形態3による記憶データ保護装置におけるアクセス権管理テーブル更新処理の流れを示すフローチャートである。

【図10】 この発明の実施の形態4による記憶データ保護装置を示すブロック図である。

【図11】 実施の形態4による記憶データ保護装置におけるモジュールインストール処理の流れを示すフロー

チャートである。

【図 12】 メモリ獲得解放処理の流れを示すフローチャートである。

【図 13】 データアクセス宣言処理の流れを示すフローチャートである。

【図 14】 この発明の実施の形態 5 による記憶データ保護装置を示すブロック図である。

【図 15】 実施の形態 5 による記憶データ保護装置におけるトラップ検出処理の流れを示すフローチャートである。

【図 16】 不当アクセス内容記録処理の流れを示すフローチャートである。

【図 17】 この発明の実施の形態 6 による記憶データ保護装置を示すブロック図である。

【図 18】 モジュール管理テーブルの構造を示す説明図である。

【図 19】 実施の形態 6 による記憶データ保護装置におけるトラップ検出処理の流れを示すフローチャートで

ある。

【図 20】 モジュールアンインストール処理の流れを示すフローチャートである。

【図 21】 従来の記憶データ保護装置を示すブロック図である。

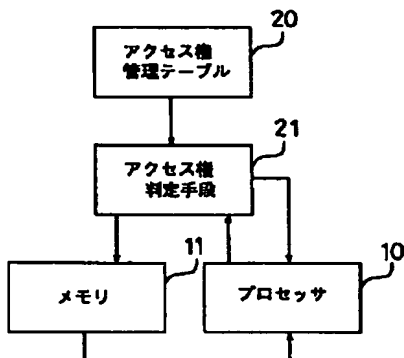
【図 22】 アクセス権管理テーブルの構造を示す説明図である。

【図 23】 記憶データ保護装置におけるアクセス権判定処理の流れを示すフローチャートである。

#### 10 【符号の説明】

10 プロセッサ、11 メモリ、20 アクセス権管理テーブル、21 アクセス権判定手段、30 モジュール管理テーブル、31 データセグメント管理テーブル、40 アクセス権管理テーブル更新手段、50 モジュールインストール手段、51 メモリ獲得解放手段、52 データアクセス宣言手段、60 二次記憶装置、61 トラップ検出手段、62 不当アクセス内容記録手段、70 モジュールアンインストール手段。

【図 1】



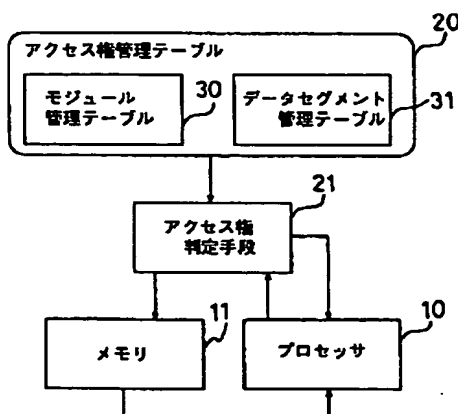
【図 2】

15 データセグメント インデックス	16 データセグメント 開始アドレス	17 データセグメント サイズ	22 アクセス可能モジュール 開始アドレス	23 アクセス可能モジュール サイズ	20
1	0x10000	0x500	0x000	0x15000	
2					
⋮	⋮	⋮	⋮	⋮	
n					

【図 5】

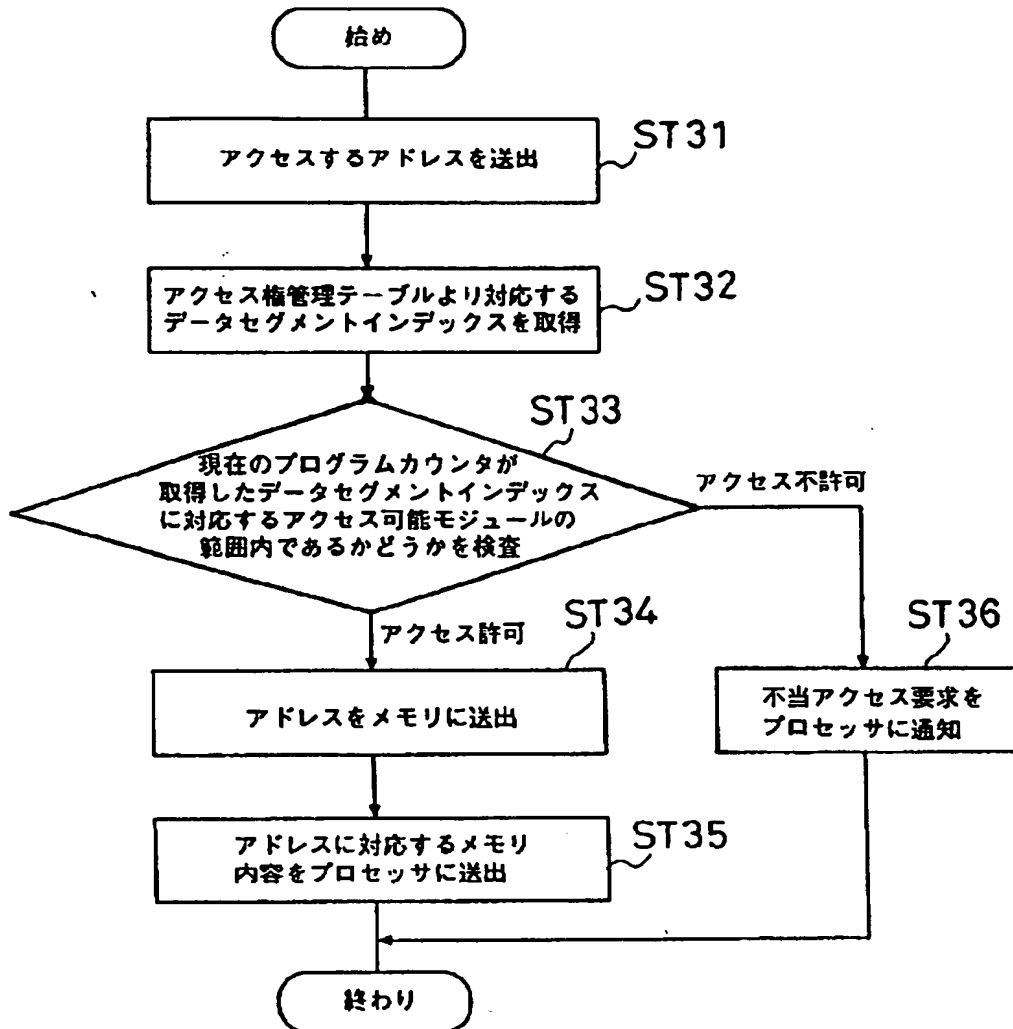
32 モジュール インデックス	33 モジュール 開始アドレス	34 モジュール サイズ	30
1	0x000	0x15000	
2	0x10000	0x1000	
⋮	⋮	⋮	
n			

【図 4】





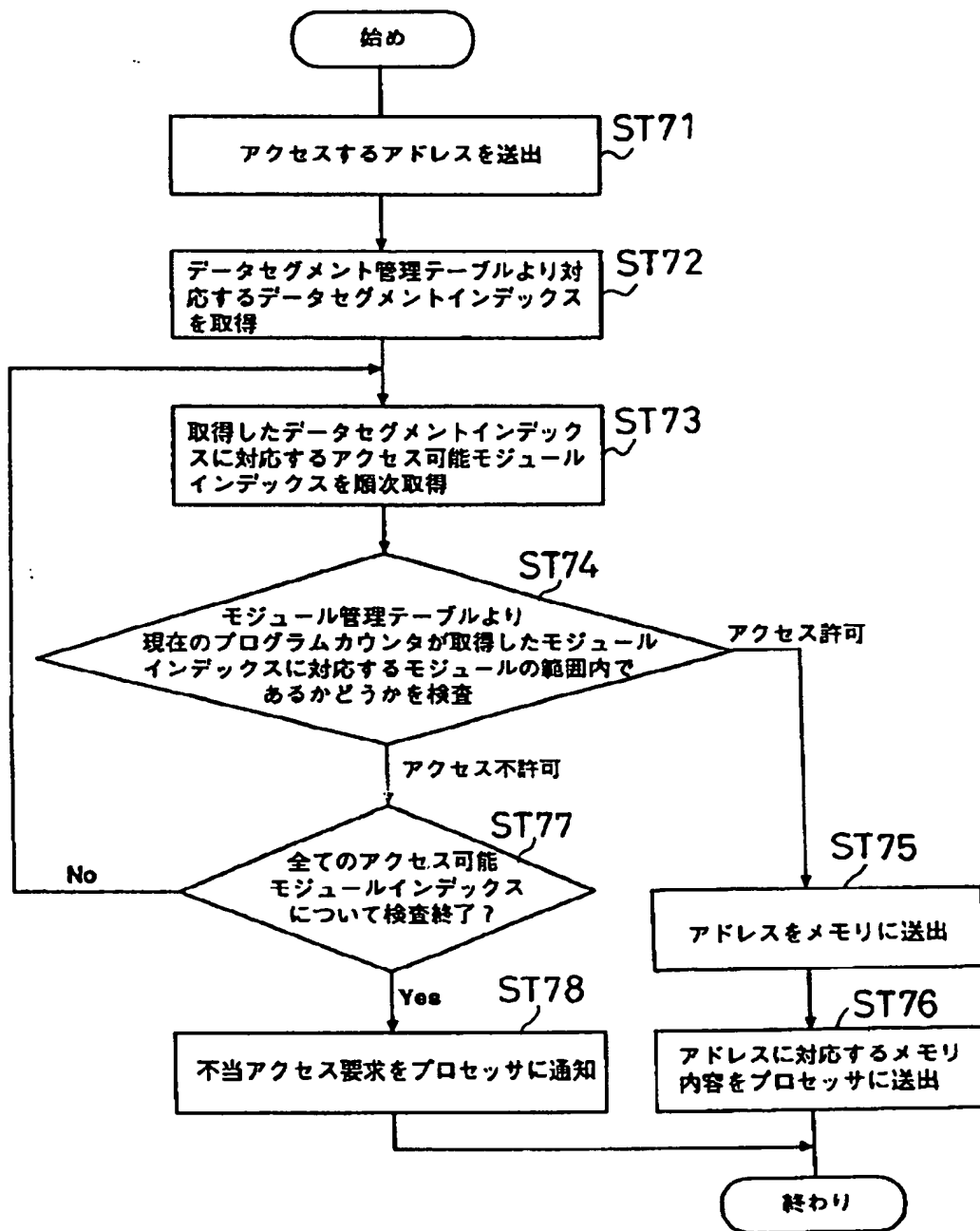
【図3】



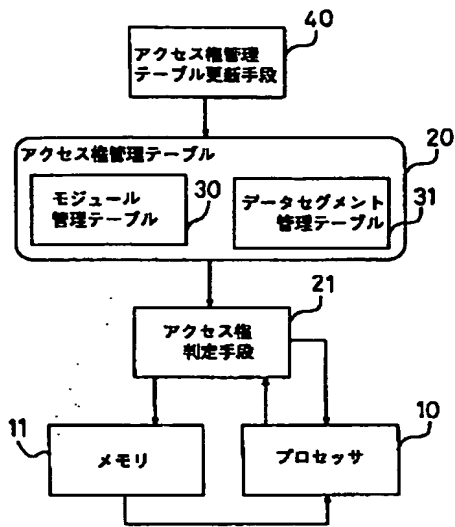
【図6】

15 データセグメント インデックス	16 データセグメント 開始アドレス	17 データセグメント サイズ	35 アクセス可能 モジュールインデックス	31
1	0xa10000	0x500	1, 2, 3, ...	
2	0xa10600	0x1000	2, 5	
⋮	⋮	⋮	⋮	
n	⋮	⋮	⋮	

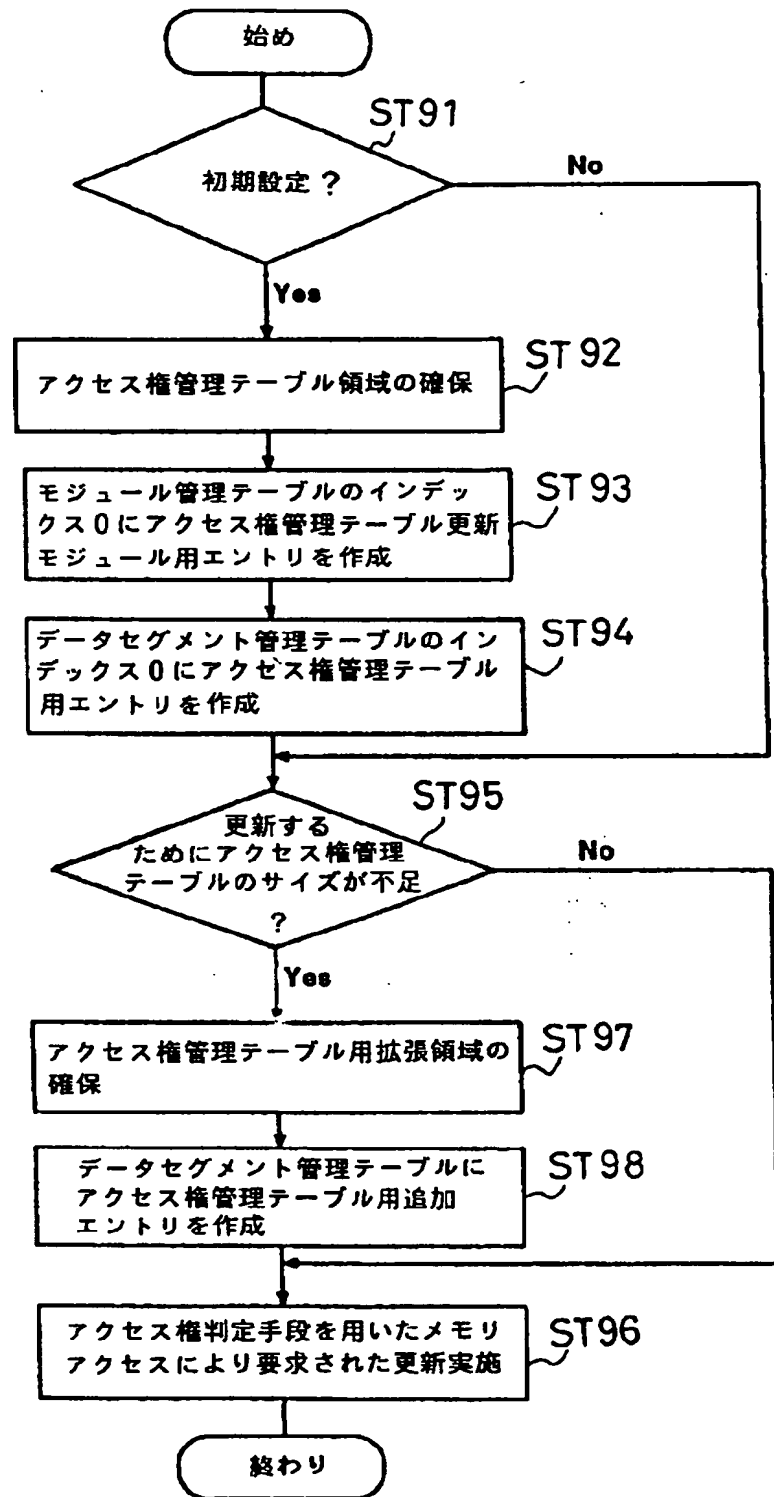
【図7】



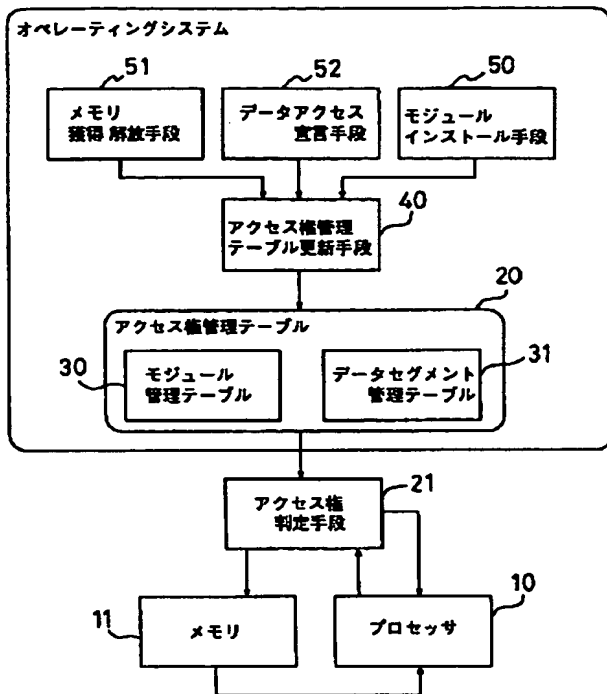
【図8】



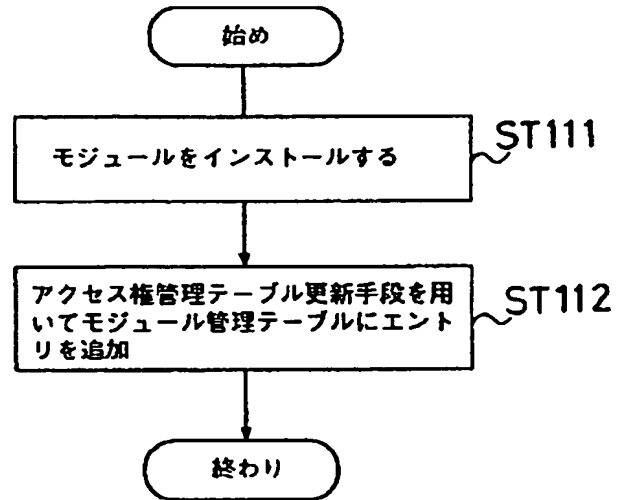
【図9】



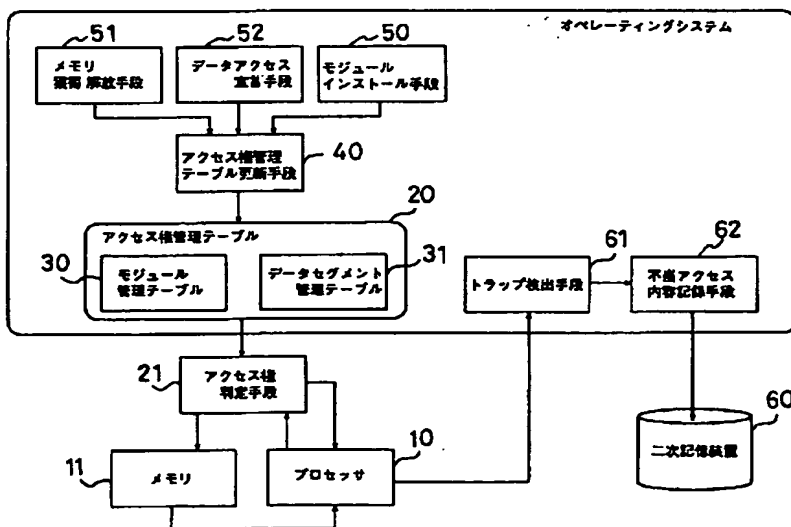
【図 10】



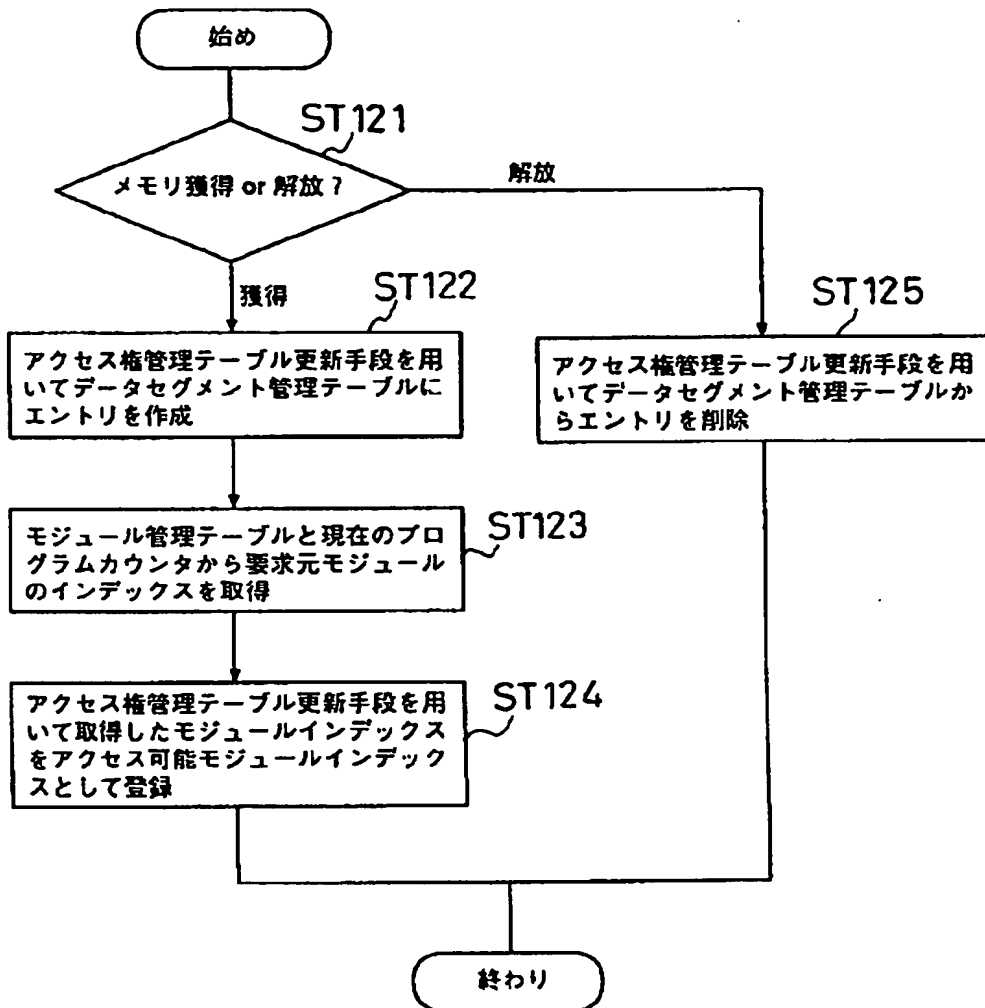
【図 11】



【図 14】



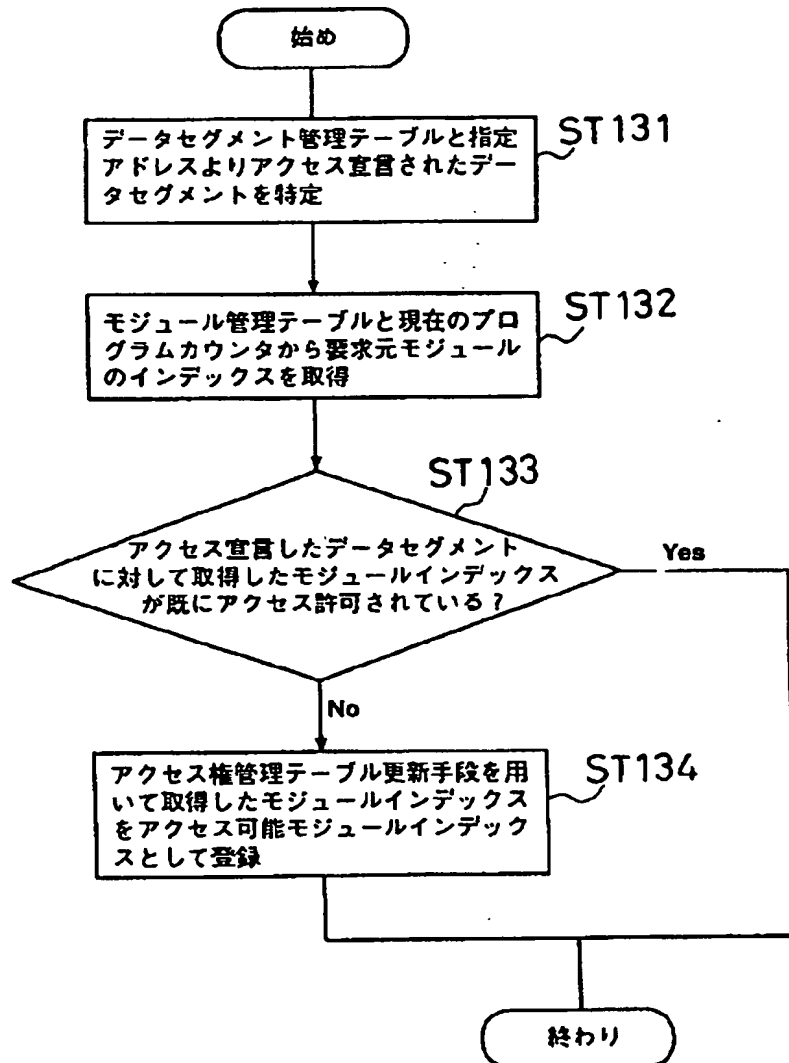
【図 12】



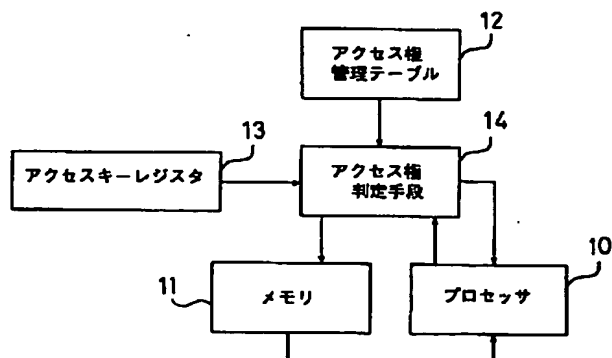
【図 18】

32 モジュール インデックス	33 モジュール 開始アドレス	34 モジュール サイズ	71 切り離し 可能フラグ	72 接続状態	73 切り離し手続き 開始アドレス	30
0: 予約 [アクセス権管理テーブル更新モジュール]	0x1dc000	0x2000	×	イン ストール	——	
1	0xa000	0x15000	○	イン ストール	0x13000	
2	0x1f000	0x1000	○	アンイン ストール	0x1fe00	
⋮	⋮	⋮	⋮	⋮	⋮	
n	⋮	⋮	⋮	⋮	⋮	

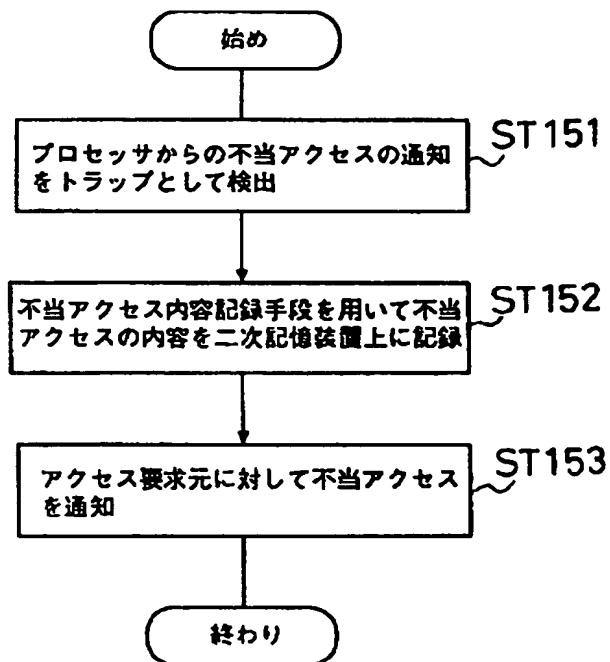
【図 1 3】



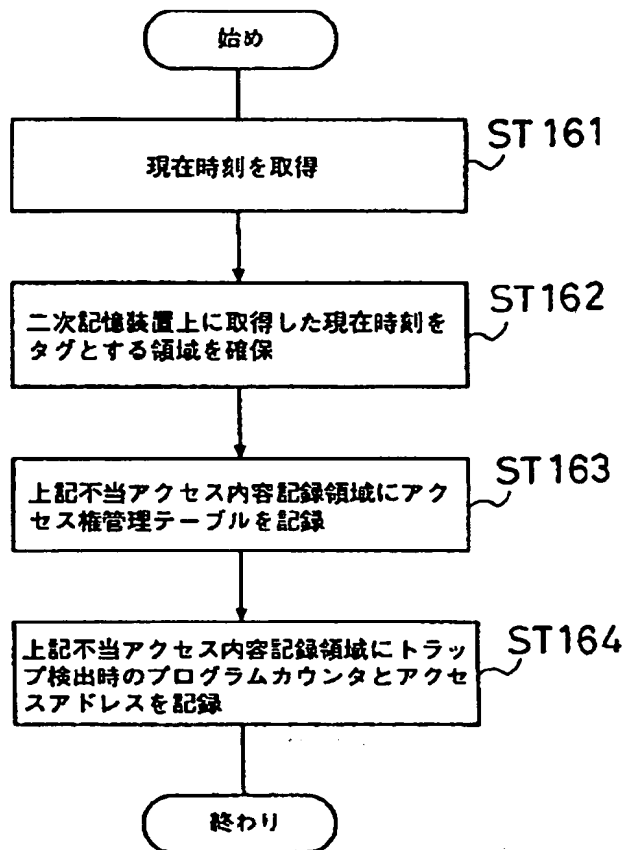
【図 2 1】



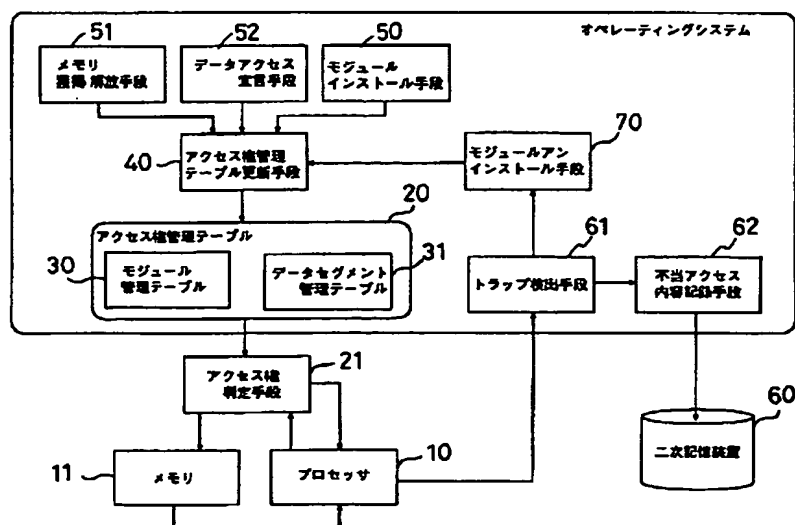
【図 15】



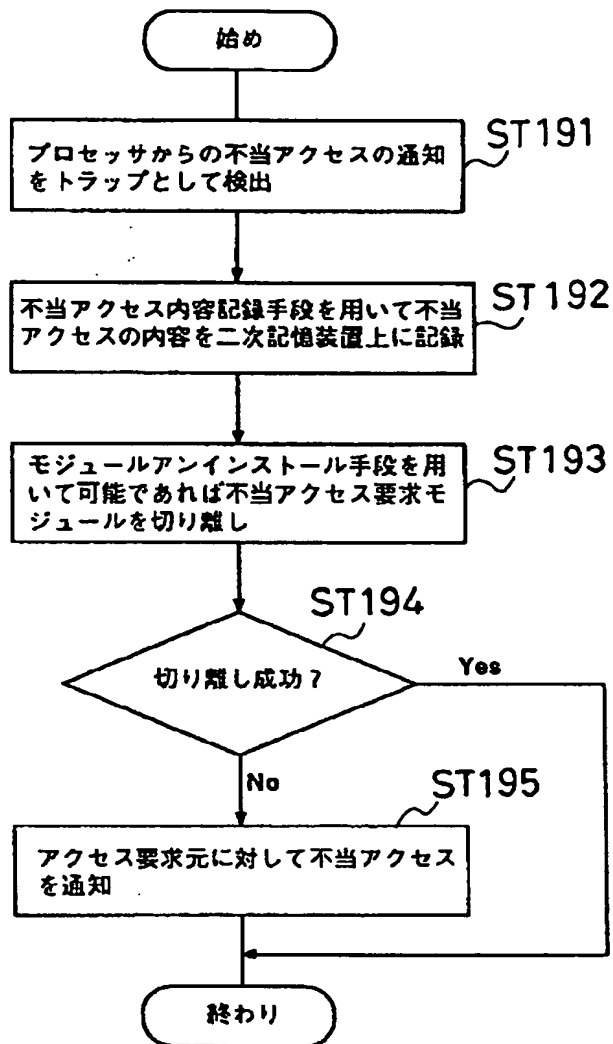
【図 16】



【図 17】



【図 19】

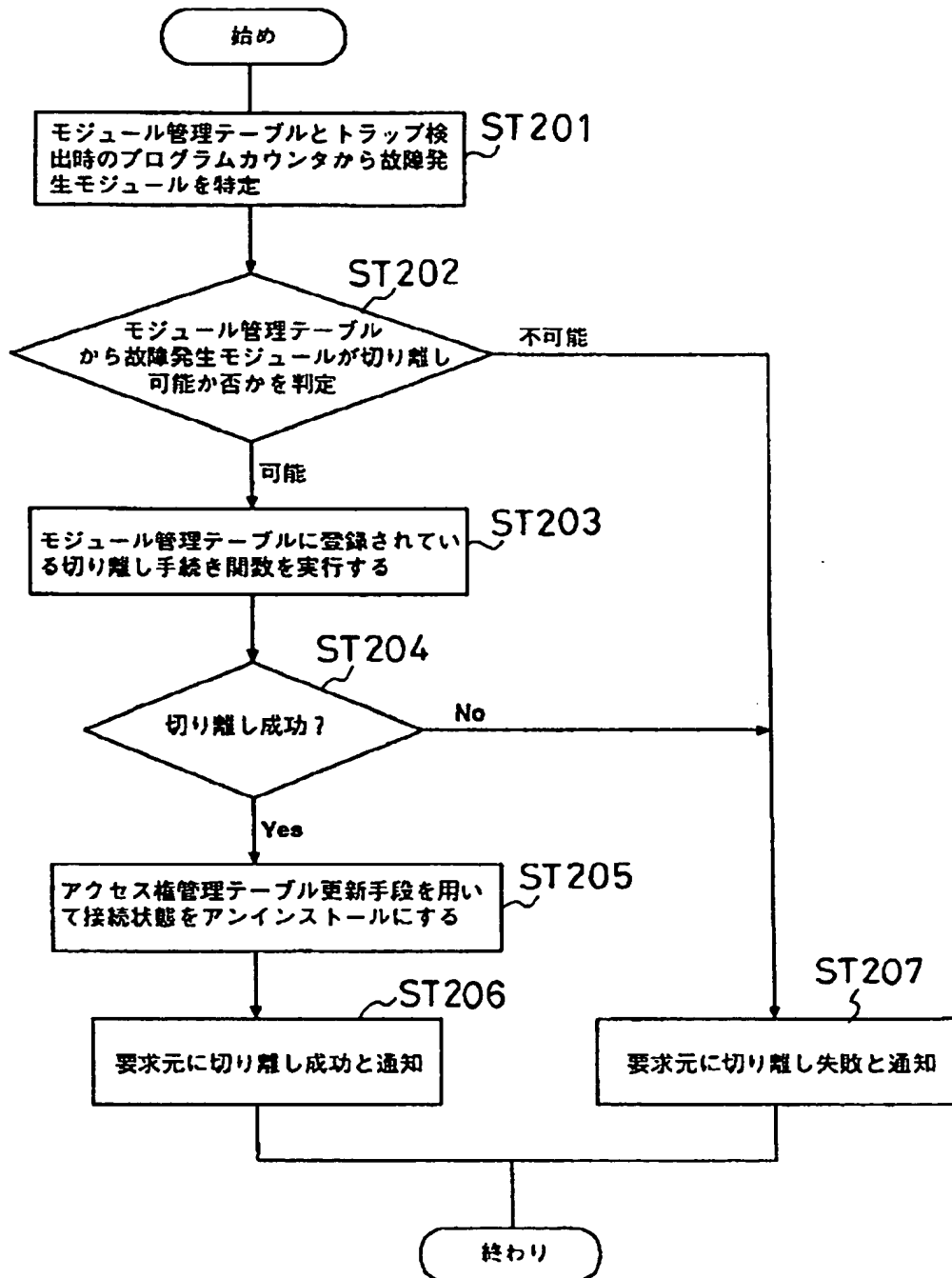


【図 22】

15 データセグメント インデックス	16 データセグメント 開始アドレス	17 データセグメント サイズ	18 メモリアクセスキー	12
1	0xa10000	0x600	32	
2	⋮	⋮	⋮	
⋮	⋮	⋮	⋮	
n	⋮	⋮	⋮	



【図20】



【図23】

